

**In the Claims:**

Please amend claims 1, 22, 44-58 and 60-64 as indicated below.

1. (currently amended) A thread-safe debugging system, comprising:
  - a thread-safe debug service which is executable on a client computer system to provide debug services to debug a multi-threaded application which is executable on the client computer system; and
  - a thread-safe remote control service which is executable on the client computer system to receive control requests from an external source to initiate and manage the debug services on the client computer system after initiation of the multi-threaded application, wherein the external source comprises a remote computer system external to the client computer system.
2. (original) The thread-safe debugging system of claim 1, wherein the debug client comprises a thread-safe debug print function which is operable independently of the remote control service, wherein the thread-safe debug print function is operable to provide debug output for one or more threads of the multi-threaded application such that the debug output of each of the one or more threads remains distinct from the debug output of the other threads.
3. (original) The thread-safe debugging system of claim 2, wherein the debug output of each of the one or more threads may be directed to an output target.
4. (original) The thread-safe debugging system of claim 3, wherein the output target comprises a file.
5. (original) The thread-safe debugging system of claim 3, wherein the output target comprises a standard output terminal.

6. (original) The thread-safe debugging system of claim 3, wherein the output target comprises a recipient computer system coupled to the client computer system.
7. (original) The thread-safe debugging system of claim 3, wherein the output target comprises a plurality of remote diagnostic tools.
8. (previously presented) The thread-safe debugging system of claim 2, wherein the thread-safe debug print function is operable to provide debug output in a plurality of regional or national languages.
9. (original) The thread-safe debugging system of claim 1, wherein the remote control service is operable to switch the debug services on and off after initiation of the multi-threaded application.
10. (original) The thread-safe debugging system of claim 1, wherein the debug service comprises one or more debug objects, wherein each of the one or more debug objects corresponds to a component of the multi-threaded application, wherein each of the one or more debug objects is operable to provide the debug services to the corresponding component of the multi-threaded application.
11. (original) The thread-safe debugging system of claim 10, wherein the remote control service is operable to allow a remote source to switch the debug services on and off for each of the corresponding components of the multi-threaded application by referencing each of the corresponding debug objects by name.
12. (original) The thread-safe debugging system of claim 10, wherein the remote control service is operable to allow a remote source to switch the debug services on and off for a set of the corresponding components of the multi-threaded application by specifying a pattern to select a set of the debug objects by name.

13. (original) The thread-safe debugging system of claim 10, wherein the debug services comprise a service to list all the debug objects in the multi-threaded application.
14. (original) The thread-safe debugging system of claim 10, wherein the debug services comprise a service to list a state of each debug object in the multi-threaded application.
15. (original) The thread-safe debugging system of claim 1, wherein the debug services comprise a service to profile execution of the multi-threaded application.
16. (canceled)
17. (original) The thread-safe debugging system of claim 1, wherein the debug services comprise a service to trace program execution through the multi-threaded application.
18. (original) The thread-safe debugging system of claim 1, wherein the debug services comprise a service to collect run-time statistics on the execution of the multi-threaded application.
19. (original) The thread-safe debugging system of claim 1, wherein the debug services comprise a service to log statistical information on the execution of the multi-threaded application.
20. (original) The thread-safe debugging system of claim 1, wherein the debug services comprise a service to log performance information on the execution of the multi-threaded application.
21. (previously presented) The thread-safe debugging system of claim 1, wherein the external source comprises a third party application executing on a remote computer

system, wherein the remote computer system is coupled to the client computer system over a network.

22. (currently amended) A thread-safe debugging method, comprising:

initiating execution of a multi-threaded application on a client computer system, wherein the multi-threaded application is linked to a thread-safe debug service, and wherein the debug service is executable to provide debug services to debug the multi-threaded application;

initiating one of the debug services from a remote source through a thread-safe remote control service executing on the client computer system after initiating execution of the multi-threaded application on the client computer system, wherein the remote control service comprises a network protocol interface which is operable to receive control requests from the remote source, wherein the remote source comprises a remote computer system external to the client computer system; and

managing the one of the debug services from the remote source through the remote control service after initiating the debug service.

23. (original) The thread-safe debugging method of claim 22, wherein the thread-safe debug service comprises a thread-safe debug print function which is operable independently of the remote control service, wherein the thread-safe debug print function is operable to provide debug output for one or more threads of the multi-threaded application such that the debug output of each of the one or more threads remains distinct from the debug output of the other threads.

24. (original) The thread-safe debugging method of claim 23, wherein the debug output of each of the one or more threads may be directed to an output target.

25. (original) The thread-safe debugging method of claim 24, wherein the output target comprises a file.
26. (original) The thread-safe debugging method of claim 24, wherein the output target comprises a standard output terminal.
27. (original) The thread-safe debugging method of claim 24, wherein the output target comprises a recipient computer system coupled to the client computer system.
28. (original) The thread-safe debugging method of claim 24, wherein the output target comprises a plurality of remote diagnostic tools.
29. (previously presented) The thread-safe debugging method of claim 23, wherein the thread-safe debug print function is operable to provide debug output in a plurality of regional or national languages.
30. (original) The thread-safe debugging method of claim 22, wherein the remote control service is operable to allow a remote source to switch the debug services on and off after initiation of the multi-threaded application.
31. (original) The thread-safe debugging method of claim 22, wherein the debug service comprises one or more debug objects, wherein each of the one or more debug objects corresponds to a component of the multi-threaded application, wherein each of the one or more debug objects is operable to provide the debug services to the corresponding component of the multi-threaded application.
32. (original) The thread-safe debugging method of claim 31, wherein the multi-threaded application is linked to a thread-safe debug service by embedding each of the one or more debug objects into the corresponding component of the multi-threaded application.

33. (original) The thread-safe debugging method of claim 31, wherein the remote control service is operable to allow a remote source to switch the debug services on and off for each of the corresponding components of the multi-threaded application by referencing each of the corresponding debug objects by name.

34. (original) The thread-safe debugging method of claim 31, wherein the remote control service is operable to allow a remote source to switch the debug services on and off for a set of the corresponding components of the multi-threaded application by specifying a pattern to select a set of the debug objects by name.

35. (original) The thread-safe debugging method of claim 31, wherein the managing the one of the debug services comprises listing all the debug objects in the multi-threaded application.

36. (original) The thread-safe debugging method of claim 31, wherein the managing the one of the debug services comprises listing a state of each debug object in the multi-threaded application.

37. (original) The thread-safe debugging method of claim 22, wherein the managing the one of the debug services comprises profiling execution of the multi-threaded application.

38. (canceled)

39. (original) The thread-safe debugging method of claim 22, wherein the managing the one of the debug services comprises tracing program execution through the multi-threaded application.

40. (original) The thread-safe debugging method of claim 22, wherein the managing the one of the debug services comprises collecting run-time statistics on the execution of the multi-threaded application.

41. (original) The thread-safe debugging method of claim 22, wherein the managing the one of the debug services comprises logging statistical information on the execution of the multi-threaded application.

42. (original) The thread-safe debugging method of claim 22, wherein the managing the one of the debug services comprises logging performance information on the execution of the multi-threaded application.

43. (previously presented) The thread-safe debugging method of claim 22, wherein the external source comprises a third party application executing on a remote computer system, wherein the remote computer system is coupled to the client computer system over a network.

44. (currently amended) A ~~carrier~~ storage medium comprising program instructions for thread-safe debugging, wherein the program instructions are computer-executable to perform:

linking a thread-safe debug service to a multi-threaded application executable on a client computer system, wherein the thread-safe debug service comprises one or more debug services for debugging the multi-threaded application;

initiating one of the debug services from a remote source through a thread-safe remote control service executing on a server computer system after initiation of the multi-threaded application, wherein the remote control service comprises a network protocol interface which is operable to receive control requests from the remote source, wherein the remote source comprises a remote computer system external to the client computer system; and

managing the one of the debug services from the remote source through the remote control service after initiating the debug service.

45. (currently amended) The ~~carrier~~ storage medium of claim 44, wherein the thread-safe debug service comprises a thread-safe debug print function which is operable independently of the remote control service, wherein the thread-safe debug print function is operable to provide debug output for one or more threads of the multi-threaded application such that the debug output of each of the one or more threads remains distinct from the debug output of the other threads.

46. (currently amended) The ~~carrier~~ storage medium of claim 45, wherein the debug output of each of the one or more threads may be directed to an output target.

47. (currently amended) The ~~carrier~~ storage medium of claim 46, wherein the output target comprises a file.

48. (currently amended) The ~~carrier~~ storage medium of claim 46, wherein the output target comprises a standard output terminal.

49. (currently amended) The ~~carrier~~ storage medium of claim 46, wherein the output target comprises a recipient computer system coupled to the client computer system.

50. (currently amended) The ~~carrier~~ storage medium of claim 46, wherein the output target comprises a plurality of remote diagnostic tools.

51. (currently amended) The ~~carrier~~ storage medium of claim 45, wherein the thread-safe debug print function is operable to provide debug output in a plurality of regional or national languages.



52. (currently amended) The ~~carrier~~ storage medium of claim 44, wherein the remote control service is operable to allow a remote source to switch the debug services on and off after initiation of the multi-threaded application.

53. (currently amended) The ~~carrier~~ storage medium of claim 44, wherein the debug service comprises one or more debug objects, wherein each of the one or more debug objects corresponds to a component of the multi-threaded application, wherein each of the one or more debug objects is operable to provide the debug services to the corresponding component of the multi-threaded application.

54. (currently amended) The ~~carrier~~ storage medium of claim 53, wherein the remote control service is operable to allow a remote source to switch the debug services on and off for each of the corresponding components of the multi-threaded application by referencing each of the corresponding debug objects by name.

55. (currently amended) The ~~carrier~~ storage medium of claim 53, wherein the remote control service is operable to allow a remote source to switch the debug services on and off for a set of the corresponding components of the multi-threaded application by specifying a pattern to select a set of the debug objects by name.

56. (currently amended) The ~~carrier~~ storage medium of claim 53, wherein the managing the one of the debug services comprises listing all the debug objects in the multi-threaded application.

57. (currently amended) The ~~carrier~~ storage medium of claim 53, wherein the managing the one of the debug services comprises listing a state of each debug object in the multi-threaded application.

58. (currently amended) The ~~carrier~~ storage medium of claim 44, wherein the managing the one of the debug services comprises profiling execution of the multi-threaded application.

59. (canceled)

60. (currently amended) The ~~carrier~~ storage medium of claim 44, wherein the managing the one of the debug services comprises tracing program execution through the multi-threaded application.

61. (currently amended) The ~~carrier~~ storage medium of claim 44, wherein the managing the one of the debug services comprises collecting run-time statistics on the execution of the multi-threaded application.

62. (currently amended) The ~~carrier~~ storage medium of claim 44, wherein the managing the one of the debug services comprises logging statistical information on the execution of the multi-threaded application.

63. (currently amended) The ~~carrier~~ storage medium of claim 44, wherein the managing the one of the debug services comprises logging performance information on the execution of the multi-threaded application.

64. (currently amended) The ~~carrier~~ storage medium of claim 44, wherein the external source comprises a third party application executing on a remote computer system, wherein the remote computer system is coupled to the client computer system over a network.